BASICS OF ANIMATION

Basics of Animation is a set of simple programs designed to teach those new to computers how to make shapes appear to move around on the screen. The three methods demonstrated are animation using the PRINT COMMAND, PLOT COMMAND, and ATARI'S great PLAYER/MISSILE GRAPHICS.

The person using this lesson should be familiar with BASIC programming so that he or she can read the code that is included. Since the program is not protected, the user is encouraged to try their own modifications inorder to gain a greater perspective of the art of animation.

SANTA CRUZ

EDUCATIONAL SOFTWARE

presents

BASICS of ANIMATION

TRICKY TUTORIAL tm

#4

(c) 1981 by S.C.E.S.

HOW TO LOAD
TRICKY TUTORIALS

TAPE: First,if you haven't cleaned your tape recorder heads lately,please do so now.Also,the heads should be de-magnitized every few weeks. SEE ANY STEREO SHOP FOR THE RIGHT TOOLS.

PLACE THE TAPE WITH THE LABEL SIDE UP IN YOUR RECORDER. MAKE SURE IT IS REWOUND AND ALSO RESET THE COUNTER TO 0. PUSH PLAY ON THE RECORDER. TYPE RUN"C:" AND PRESS RETURN.

IF IT DOESN'T LOAD PROPERLY TRY MOVING YOUR TAPE FORWARD A LITTLE BIT SO THAT THE NUMBER STARTS AT 1(OR IF NECESSARY 2,3,...) . BY GRADUALLY POSITIONING THE TAPE FORWARD YOU SHOULD FIND A POSITION WHERE IT WILL LOAD. WE OCCASIONALLY SHIP A BAD TAPE OR DISK (OUR FIRST SUPPLIER OF TAPES SOLD US JUNK). IF THIS HAPPENDS CALL (408) 476-4901 AND REQUEST ANOTHER.

Once the program starts it will load in, in several parts.This is to allow you to run it on minimal machines of 16k of memory. If you have less than this, why not add some more. You can now buy memory very cheap. With more memory you can go beyond these simple tutorials. After each part, the computer will beep. This is your signal to press return to load in the next part. Most programs will run themselves when done loading.

DISK: To load & run the disk you first have to turn on the drive. When the busy light goes out place the disk in the drive. Now turn on the computer with the basic cartridge in place and the program will load each part and run by itself (aren't disk drives nice!). You must have 24k of memory, but since a computer with a drive and 16k has almost no program room we assume you have this much.

NOTE:MANY OF THE EARLY DRIVES HAD A LOT OF PROBLEMS WITH HEAD ALIGNMENT AND MOTOR SPEED. NEWER DRIVES OR THOSE(LIKE OURS) WHICH HAVE BEEN REBUILT RECIENTLY BY ATARI WILL LOAD WITHOUT TROUBLE. IF YOU DO HAVE PROBLEMS LOADING TRY SEVERAL STORES OR FRIENDS DRIVES TO BE SURE THE PROBLEM IS NOT YOURS. IF OUR DISKS ARE DEFECTIVE WE WILL REPLACE THEM.

Introduction To Santa Cruz Software

Hi! First let us thank you for purchasing our program. We want to start out by telling you a little about ourselves and our company. Santa Cruz Educational Software is the result of several local programmers who initially purchased the ATARI computer as a home machine around the time the 800 was first sold. At that time, and even today, we found a lack of information about how to use the power of the machine. As you have seen in a few of the better programs being sold, the ATARI can do more than any other computer in it's price range.

The people at ATARI have been more supportive than any other small computer company that we know of. In early 1981 they released the Operating System and Hardware Manuals, and in the spring they allowed us to attend a seminar to explain the machine to vendors, but.....for the average user of the machine these sources of information are not understandable. For this reason, we saw the need to write a series of simple programs that would explain these more advanced features to the general public. This, then, is what the Tricky Turorials(tm) are. We also sell programs that others have written and give about the highest royalties in the industry (send us yours and we'll call back to discuss their sale!

Write us so that we can know what you want explained in the tutorial series. Even though we are very busy , we usually answer serious mail. If you have trouble finding equiptment or live in a remote area, we can get it for you. We are here to help you.

Finally, please remember that this is a part time business and unless you support us with suggestions, comments, and of course orders, we won't survive. We get the bugs out as fast as you tell us about them!

WRITE US WITH ANY BUGS YOU FIND AND THE SOLUTION TO THE BUG AND WE WILL SEND YOU SOME FREE SOFTWARE! WE WANT TO BE THE PEOPLES COMPUTING COMPANY!!

BASICS OF ANIMATION

What does animation mean to you? Were you hoping that by purchasing this program you would then be able to do cartoons on the screen in intricate detail...or did you perhaps just hope to learn some basics to move a few shapes around the screen for a game or business application you are programming? Well, we're going to start out this lesson by discussing what can and also cannot be done on your ATARI.

Animation requires two main qualities to appear nicely on your screen. First you must control enough points so that something seems to be happening and then you must move these points around fast enough so that they apear lifelike.

Rule number one:
IT CAN'T BE DONE FROM BASIC

Wait! Don't panic and return this program to us before you read on. We know your not a assembly language programmer . Neither are we except when we have to.

Basic in the ATARI is slow because it is an interpeter, meaning that every time you tell it to go move some point around the screen it has to first go and figure out exactly how to do what you've asked. This takes so much time that even with the machine doing hundreds of thousands of steps per SECOND, it can only move a single point around as fast as our first two examples. The way around this is to do one of four things:

1)You can program in themachines language where it doesn't have to interpet. This we already rejected asto difficult for the average ATARI user.

2)You can use machine language routines built into the machine. This is what we do in part two using the print command to draw complicated shapes very quickly. In fact if you look at the code for the space bug you will see adelay loop was put in to slow it down! Also, although too complicated to explain in a basic program,you can redefine the characters you are printing to draw almost anything you can imagine. The monsters in SPACE INVADERS are redefined letters of the alphabet, and are moved about by print commands. There are a few programs in the $40.00 range for sale that will move compliated shapes around the screen fast(U.S.A. software has an excellant one).

3)You could also go buy a better basic that includes some machine language movement routines to move your shapes around.(Basic A+ or Microsoft Basic).

4)You can use PLAYER/MISSILE GRAPHICS like the tie fighter program we include as an example.

We now explain each part of our program:

ANIMATION USING THE PLOT COMMAND

If you haven't already done so, now is the time to load in the program. Follow the instructions in the program to use your joystick to move a simple square around the screen. This is similiar to the movement in games of the SURROUND type. As you can see while you follow the program as it shows you the code, it is really quite simple to move a square! The better games of this type add another player and leave a controlled trail that you have the computer keep tract of. Then when you touch a location where the computer's "records" show a trail is, it scores for the other guy.

After you have run the program through to the point where we show you how to erase the trail, we suggest you stop the program (press stop and rewind the tape or remove the disk). Now press reset or break to stop the program and type new. You can always come back and finish it later. For disk just type RUN"D:PLOT2.DSK" to continue. The reason for stoping is for you to now write a small program that moves a square around using the plot statement. The easiest way would be for you to look at the listing at the end of this manual. Lines numbered from 10520 to 10610 are the ones that move the square around. Add to our program using your own ideas. Also, since the code exists in the program we sent you, please feel free to delete the lines you don't need, and start with them already in the computer to save time(the assembler cartridge could do this quickly!).Please be sure if you save your work it is on a separate tape or disk. Even if you intend to do only non-game applications, the learning you gain from writing a simple game will help other attempts at animation.

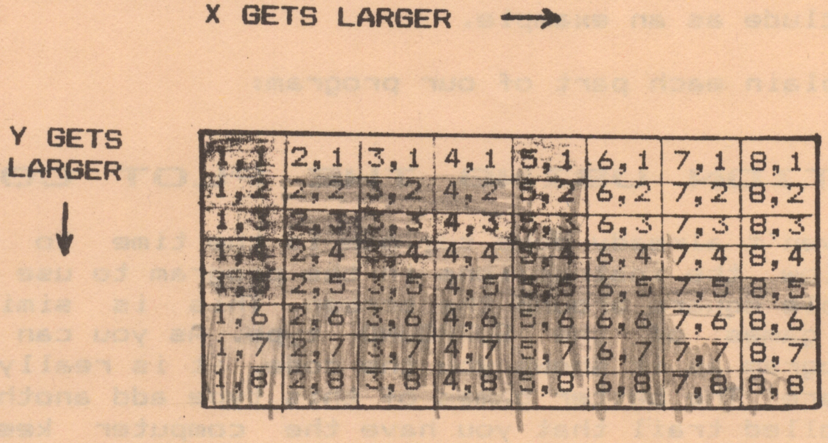
Before going on, we want to point out that when using plot and erasing your "trail", you have to actually plot the position you are erasing with the color of the background. This makes that position seem to dissapear as you plot the next position in the color of your choice.

The next thing we do to show you the basics of animation is to draw out a tie fighter.

PLEASE NOTE THAT THIS IS ONLY AN EXAMPLE AND THE SHAPE COULD BE ANYTHING FROM A BUSINESS TO A HOME APPLICATION.IT ALSO COULD BE MUCH LARGER THAN WE DREW.

Here is how to input points to plot:

BASICS OF ANIMATION



1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1
1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2
1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3
1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4
1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5
1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6
1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7
1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8

FIG.1.

Look at fig.1. We drew out a shape at the upper left corner of what is called an X-Y coordinate system. The upper left most point that is filled in is at x=1 and y=1. Moving down the figure the next point is at x=1,y=2, then x=1,y=3, etc. with the last point of OUR shape at x=5,y=5. The computer doesn't care what order you input points into plot statements just so long as you input them all. This means you don't have to use the points in the same order as we did. Now, to put these into a BASIC program we use the DATA statement (read your BASIC manual!) it looks as follows:

```
DATA 1,1,2,1,3,1,4,1,5,1,2,3,3,2,
3,3,3,4,4,1,5,1,5,2,5,3,5,4,5,5
```

EVERY TWO NUMBERS REPRESENTS ONE POINT TO PLOT

If you haven't already,please go run this part of the program. The last part of this lesson allows you to move the cursor up to the data line for the tie fighter shape and enter (by typing new numbers) it back into the computers memory. If a mistake occurs reloading this part will fix it up. This is needed because the data statement must be corrected. Also, you may use numbers bigger than 8,8 for your shape, but when you move it, if it goes off screen it will bomb the program since we only allowed in our code for the smaller shape. Again, feel free to copy and modify our program or, if you like, just slowly change a small part of it to see how your own changes can add to it or make it better!!!

ANIMATION USING THE PRINT COMMAND

We include the "Bird at the Ocean " from the ATARI BASIC manual as a convenience for those who never typed it in. Again , after finishing this part please modify the shapes or the background to get a feel for your own ideas. Backgrounds are simply created with plot & drawto commands and the special graphics characters. This is how you will learn to do the things you want. WE next show you animation of a single character both in the horizontal and vertical direction. They both have simple sounds to demonstrate how easy it is to do. The program will stop and allow you to move the cursor up to the six lines that hold the shape for the "SPACE BUG". change (using the special graphic characters on the back of the BASIC manual or anything else you want) the shapes ONLY between the quotes. When done, move the cursor to the top line and press return until all six lines are re-entered into memory. Only one return is needed in our newer versions of this program. If you goof up clear the page and type goto 23500 and press return. If you enter the new shape correctly move to a BLANK area on the screen and type cont, then press return to see your shape move across the screen. When you have played enough with our code try adding backgrounds first before your shape is printed or perhaps try several shapes.

—Note—

TO DO FULL ANIMATION MANY ADDITIONAL TRICKS ARE TAUGHT IN OUR OTHER TUTORIALS. THIS IS NOT INTENDED JUST TO SELL MORE PROGRAMS. TO DO TRUE ANIMATION YOU NEED PAGE FLIPPING , MODIFICATIONS TO THE DISPLAY LIST AND SCROLLING.

PLAYER MISSILE GRAPHICS

Some of you will be mad because we neither include a listing of all the poke locations, nor explain every statement beyond the remark statements. The reason is because:

1)There are so many pokes that we wrote a memory map to give them all to you($5.95).

2)At least this is a complete example, which at the time this is being written has not been offered in the literature.

3)If you will play with our code you will learn alot. We sell a complete tutorial on Player/Missile Graphics

4)Notice we use two methods to move the missiles. This will show you the tradeoffs involved with speed.

Finally we have included an example of animating by moving color registers. We don't know who wrote this as it was submitted by a customer in Indiana. We assume it is public domain. It sure is pretty.

WE CERTAINLY HOPE YOU LEARNED
SOMETHING FROM BASICS OF ANIMATION
bye....

HERE IS THE CODE FOR THE PLAYER-MISSILE GAME

```
20 K=15:REM INITIAL POS OF P/M 1
30 TRAP 20
40 SOUND 3,13,8,2
50 PRINT "}"
60 GOSUB 680:REM BACKGROUND SUBROUTINE
70 SETCOLOR 2,0,0:X=120:Y=90:REM SET BACKGROUND COLOR AND PLAYER POSITION
80 POKE 704,133:REM SETCOLOR PLAYER0 TO BLUE
90 POKE 705,198:REM SETCOLOR PLAYER1 TO GREEN
100 A=PEEK(106)-8:POKE 54279,A:PMBASE=256*A:REM SET PLAYER-MISSLE STARTING ADDRESS
110 POKE 559,46:POKE 53277,3:REM ENABLE PM GRAPHICS WITH 2-LINE RESOLUTION
120 POKE 752,1:REM MAKE CURSOR INVISABLE
130 FOR I=PMBASE+512 TO PMBASE+640:POKE I,0:NEXT I:REM CLEAR OUT PLAYER0 FIRST
140 FOR I=PMBASE+640 TO PMBASE+768:POKE I,0:NEXT I:REM CLEAR OUT PLAYER1 FIRST.
150 FOR I=PMBASE+640+K TO PMBASE+644+K:READ B:POKE I,B:NEXT I:REM DRAW PLAYER1
160 DATA 153,189,255,189,153
170 REM DATA FOR TIE SHAPE
180 FOR I=PMBASE+512+Y TO PMBASE+516+Y:READ A:POKE I,A:NEXT I:REM DRAW PLAYER0
190 POKE 53256,0:POKE 53260,18:REM SIZE OF PLAYER0 AND ALL MISSLES
200 POKE 53257,0:REM SIZE OF PLAYER1
210 DATA 153,189,255,189,153
220 POKE 656,3:? "PRESS 1 TO STOP GAME"
230 REM SAME SHAPE FOR OTHER PLAYER
240 REM NOW COMES THE MOTION/COLLISION ROUTINES
250 IF PEEK(764)=31 THEN 1650
260 F=PTRIG(0):REM READ TRIGGERS
270 G=PTRIG(1)
280 IF F=0 OR MIS=1 THEN GOSUB 390:REM SLOW MISSLE ROUTINE
290 IF G=0 THEN GOSUB 520:REM FAST MISSLE MOVE ROUTINE
300 POKE 656,0:IF FS=1 OR GS=1 THEN ? "GREENS SCORE=";SCOREG;"
    BLUES SCORE=";SCOREB
310 FS=0:GS=0:REM THESE TELL ATARI TO PRINT SCORE
320 POKE 53278,0:REM CLEAR OUT COLLISION REGISTERS TO USE AGAIN
330 A=PADDLE(0)
340 B=PADDLE(1)
350 POKE 53248,A:IF MIS=0 AND A<>0 THEN POKE 53252,A-1:REM MOVE PLAYER0(AND MISSLE0)
360 POKE 53249,B:POKE 53253,B-1:REM SAME FOR PLAYER1
370 GOTO 250:REM GO READ PADDLES AGAIN
```

```
380 REM SOUBROUTINE FOR BLUE  MISSILE
390 SOUND 0,227-V*2,8,INT(16-V/10):REM MAKE SOUND DECREASE AS MISSILE MOVES
400 V=V+1:POKE PMBASE+384+Y-V,0:POKE PMBASE+383+Y-V,1
410 E=PEEK(53256):IF E>0 THEN HIT=1:REM TEST FOR HIT!
420 IF V>95 THEN 500:REM MISSILE MOVED FAR ENOUGH
430 MIS=1:RETURN
440 HIT=0:FS=1:REM SAYS TO PRINT NEW SCORE,MIS & HIT ARE COUNTERS
450 SOUND 1,221,10,12:POKE 53249,250:FOR W=1 TO 200:NEXT W
460 SOUND 1,0,0,0
470 POKE 53249,120:POKE 53278,0:REM REPOSITION PLAYER1 AND CLEAR COLLISION REGISTER
480 SCOREB=SCOREB+1
490 V=0:MIS=0:SOUND 0,0,0,0:RETURN
500 IF HIT=1 THEN 440
510 V=0:MIS=0:SOUND 0,0,0,0:RETURN
520 SOUND 0,200,8,8:P=0
530 FOR I=15 TO 130:POKE PMBASE+383+I,0:POKE PMBASE+384+I,4:P=PEEK(53257)
540 IF P=3 THEN 590
550 NEXT I
560 SOUND 0,0,0,0
570 RETURN
580 REM P=PEEK(53257):IF P=2 THEN RETURN
590 SOUND 0,0,0,0:GS=1:REM SAYS TO PRINT NEW SCORE
600 RESTORE 630:SOUND 1,121,8,12
610 FOR I=PMBASE+512 TO PMBASE+640:POKE I,0:NEXT I:REM CLEAR OUT PLAYER FIRST
620 Y=75:FOR I=PMBASE+512+K+Y TO PMBASE+516+K+Y:READ B:POKE I,B:NEXT I:REM DRAW PLAYER
630 DATA 153,189,255,189,153
640 FOR I=384 TO 512:POKE PMBASE+I,0:NEXT I
650 SOUND 1,0,0,0
660 SCOREG=SCOREG+1
670 RETURN
680 GRAPHICS 8:CNT=0
690 COLOR 1:X=RND(0)*319:Y=RND(0)*159:CNT=CNT+1:PLOT X,Y:IF CNT>150 THEN RETURN
700 GO TO 690
```

AND HERE IS THE CODE FOR THE SPACE BUG

```
21310 GRAPHICS 18:? #6;"NOW LETS TRY A SPACE BUG":FOR W=1 TO 600:NEXT W
21320 TRAP 23512
21330 ? "┳":X=1:Y=10:POKE 752,1:CNT=0
21340 POSITION X,Y
21345 ? "  ":POSITION X,Y+1
21350 ? "V":POSITION X,Y+2
21360 ? "▲" :POSITION X,Y+3

21370 ? "||"
21391 SOUND 1,210,10,CNT+2
21392 FOR W=1 TO 20:NEXT W:POSITION X,Y
21393 ? "  ":POSITION X,Y+1
21394 ? "  ":POSITION X,Y+2
21395 ? "  ":POSITION X,Y+3
21396 ? "  "
21397 X=X+2:POSITION X,Y
21400 ? "  ":POSITION X,Y+1
21410 ? "V" :POSITION X,Y+2
21420 ? "▲" :POSITION X,Y+3

21430 ? "\\"
21431 FOR W=1 TO 20:NEXT W:POSITION X,Y
21432 ? "  ":POSITION X,Y+1
21433 ? "  ":POSITION X,Y+2
21434 ? "  ":POSITION X,Y+3
21435 ? "  "
21440 SOUND 1,160,10,CNT+4:X=X+2:IF X>37 THEN 21460
21450 GOTO 21340
21460 X=1:CNT=CNT+1:IF CNT=5 THEN 21480
21470 GOTO 21340
21480 SOUND 1,0,0,0:
```

THIS IS HOW TO ANIMATE A SQUARE

```
10520 V=STICK(0)
10525 SETCOLOR 2,0,0:COLOR 2
10530 IF V=14 THEN Y=Y-1:PLOT X,Y+1
10540 IF V=13 THEN Y=Y+1:PLOT X,Y-1
10550 IF V=11 THEN X=X-1:PLOT X+1,Y
10560 IF V=7 THEN X=X+1:PLOT X-1,Y
10575 IF X<1 THEN X=1
10586 IF X>79 THEN X=79
10590 IF Y<1 THEN Y=1
10600 IF Y>39 THEN Y=39
10610 COLOR 1:PLOT X,Y:IF PEEK(764)=255 THEN 10520
```

AND A TIE FIGHTER

```
11820 GRAPHICS 4:X=40:Y=20:COLOR 1:GOSUB 11930:? "USE JOYSTICK TO MOVE FIGHTER"
11825 POKE 764,255:SPEED=5:SETCOLOR 2,0,0:TRAP 11820:S=SPEED
11830 V=STICK(0):IF V=15 THEN 11832
11831 GOTO 11835
11832 IF PEEK(764)=255 THEN 11830
11835 COLOR 2:GOSUB 11930
11841 Y=Y+(V=13)*S+(V=9)*S+(V=5)*S-(V=10)*S-(V=14)*S-(V=6)*S
11842 X=X-(V=10)*S-(V=11)*S-(V=9)*S+(V=6)*S+(V=7)*S+(V=5)*S
11920 COLOR 1:GOSUB 11930:IF PEEK(764)=255 THEN 11830
11930 RESTORE 11960:POKE 752,1
11940 READ A,B:IF A=0 THEN 11970
11950 PLOT A+X,B+Y:GOTO 11940
11960 DATA 1,1,1,2,1,3,1,4.1,5,2,3,3,2,3,3,3,4,4,2,4,3,4,4,5,3,6,1,6,2,6,3,6,4,6,5,0,0,0
11970 RETURN
```